The part of the code to optimize is:

v, x\_opt\_temp, sol\_status, MIPGap = gurobi\_muSigma\_sample(NN, pp, mu, sigma, q, config.timelimit, muy\_grid, sigma2y\_grid, profits\_grid2)

Please take other parts of the code as given.

The problem that I am solving is a combinatorial programming problem. Here’s the mathematical structure:

The choice variable is m=[m1, m2, m3, …, mN, m1m2, m1m3, …, m1mN, m2m3, m2m4, …, m3m4, …, mN-1mN], i.e., .

Essentially, only m1, …, mN are variables to choose.

The problem takes as given two matrices, mu153×1 and sigma153×153, and compute two intermediate variables, mux = m @ mu and sigma2x = m @ sigma @ m’. sigma is a covariance matrix so it is positive semi-definite. sigma2x is computed using Cholesky decomposition as shown in sigma2\_exp\_gen.

The objective function is a nonlinear function of mux and sigma2x. I use a tabular method: I discretize mux and sigma2x into grid points, muy\_grid and sigma2y\_grid, and compute objective function value on grid points, profits\_grid2. Then, in function gurobi\_muSigma\_sample, for a given m, I compute mux and sigma2x, then I do bilinear interpolation (i.e., using SOS2 constraints) to obtain objective function value.

I have limited research funding, so please understand that I am only able to pay based on the optimized result (run time shortened to xxx seconds), rather than based on your working hours. I have tried various ways to optimize the solution and developed some understanding. It might be efficient if you let me know what you’d like to try before you try it, just in case it is something I have already tried.

Thank you very much!